Amendments to the Claims:

      The listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

      1.   (Canceled)

      2.   (Previously presented) The method of claim 15 wherein a plurality of processors of the homogeneous multiprocessor environment are capable of executing a first instruction of a first instruction set and a second instruction of a second instruction set.

      3.   (Original) The method of claim 2 wherein the first instruction and the second instruction share an identical bit pattern but perform different operations.

      4.   (Original) The method of claim 3 wherein a first processor of the plurality of processors executes an input/output kernel program, the input/output kernel program including a first portion expressed using the first instruction set and a second portion expressed using the second instruction set.

      5.   (Original) The method of claim 3 further comprising the step of:

      converting a functional program of the functional programs expressed using the first instruction set to an equivalent functional program expressed using the second instruction set.

      6.   (Previously presented) The method of claim 3 wherein the tasks comprise:

      x86 processing;

      graphic image processing;

      video processing;

      audio processing; and

communication processing.

7.   (Original) The method of claim 3 further comprising the step of:

receiving the initial data from a first input/output device.

8.   (Original) The method of claim 3 further comprising the steps of:

passing the resulting data to a first input/output device.

9.   (Original) The method of claim 8 wherein the step of passing the resulting data to the first input/output device further comprises the step of:

passing the resulting data through an intermediary device, wherein the intermediary device is coupled to the first input/output device and to a second input/output device.

10.   (Original) The method of claim 9 wherein the step of passing the resulting data through an intermediary device, wherein the intermediary device is coupled to the first input/output device and to a second input/output device further comprises the step of:

automatically adapting to a reallocation of the available processing resources among the tasks.

11.   (Original) The method of claim 8 wherein the step of passing the resulting data to a first input/output device further comprises the step of:

passing the resulting data to a mixed-signal device.

12.   (Original) The method of claim 3 wherein the step of allocating the available processing resources among the tasks is dynamically adjusted.

13.   (Canceled)

14.   (Canceled)

15.   (Currently amended) A method for providing multimedia functionality in a homogeneous multiprocessor environment comprising:

queuing tasks;

identifying available processing resources in the homogeneous multiprocessor environment independent of the tasks;

allocating the available processing resources among the tasks based on the capabilities of each of the available ~~processing resources~~ processors of the homogeneous multiprocessor environment and the processing requirements of each of the tasks;

providing to the available processing resources functional programs and initial data corresponding to the tasks; and

performing the tasks using the available processing resources to produce resulting data, wherein the functional programs cause the available processing resources to perform the tasks of at least one of: graphics image processing, video processing, audio processing and communications processing.

16.   (Currently amended) A method for providing multimedia functionality in a homogeneous multiprocessor environment comprising:

queuing tasks;

keeping track, remotely from the resources, of the capabilities of all available ~~processing resources~~ processors of the homogeneous multiprocessor environment;

identifying, independent of the tasks, available processing resources in the homogeneous multiprocessor environment based solely on the capabilities kept track of remotely;

allocating the available processing resources among the tasks based on the capabilities of each of the available processing resources and the processing requirements of each of the tasks;

providing to the available processing resources functional programs and initial data corresponding to the tasks; and

performing the tasks using the available processing resources to produce resulting data, wherein the functional programs cause the available processing resources to perform the tasks of at least one of: graphics image processing, video processing, audio processing and communications processing.

17.   (Previously presented) An apparatus comprising:

a plurality of processors coupled to a bus;

an input/output interface coupled to the bus;

a plurality of input/output devices coupled to the input/output interface, the plurality of processors processing program code configured to perform a plurality of tasks, the program code comprising:

program code configured to cause a first portion of the plurality of processors to interact with a first input/output device of the plurality of input/output devices;

program code configured to cause a second portion of the plurality of processors to interact with a second input/output device of the plurality of input/output devices;

program code configured to cause the second portion of the plurality of processors to emulate a specific microprocessor instruction set;

wherein the first portion of the plurality of processors provides functionality as found in a first application-specific subsystem and wherein the first input/output device is the first application-specific subsystem;

wherein the second portion of the plurality of processors provide functionality as found in a second application-specific subsystem and wherein the second input/output device is the second application-specific subsystem; and

kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors without regard to a processor mode.

18.   (Currently amended) A method for providing multimedia functionality in a homogeneous multiprocessor environment comprising:

queuing tasks;

identifying available processing resources in the homogeneous multiprocessor environment independent of the tasks;

allocating the available processing resources among the tasks based on the capabilities of each of the available ~~processing resources~~ processors of the homogeneous multiprocessor environment and the processing requirements of each of the tasks;

providing to the available processing resources functional programs corresponding to the tasks; and

performing the tasks using the available processing resources to produce resulting data wherein the functional programs cause the available processing resources to perform the tasks of at least one of graphics image processing, video processing, audio processing and communications processing.

19.   (New) The method of claim 15 wherein allocating the available processing resources among the tasks based on the capabilities of each of the available processors of the homogeneous multiprocessor environment comprises allocating the available processing resources among the tasks based on the ability of each of the available processors of the

homogeneous multiprocessor environment to be divided or aggregated with another processor of the homogeneous multiprocessor environment to provide a processing resource.

20.   (New) The method of claim 16 wherein allocating the available processing resources among the tasks based on the capabilities of each of the available processors of the homogeneous multiprocessor environment comprises allocating the available processing resources among the tasks based on the ability of each of the available processors of the homogeneous multiprocessor environment to be divided or aggregated with another processor of the homogeneous multiprocessor environment to provide a processing resource.

21.   (New) The method of claim 18 wherein allocating the available processing resources among the tasks based on the capabilities of each of the available processors of the homogeneous multiprocessor environment comprises allocating the available processing resources among the tasks based on the ability of each of the available processors of the homogeneous multiprocessor environment to be divided or aggregated with another processor of the homogeneous multiprocessor environment to provide a processing resource.